Artificial compressibility methods for the incompressible Navier-Stokes equations using lowest-order face-based schemes on polytopal meshes

Riccardo Milani, Jérôme Bonelle, Alexandre Ern

Artificial compressibility methods for the incompressible Navier–Stokes equations using lowest-order face-based schemes on polytopal meshes

Riccardo Milani[*, †] Jérôme Bonelle[*] Alexandre Ern[†]

Abstract

We investigate artificial compressibility (AC) techniques for the time discretization of the incompressible Navier–Stokes equations. The space discretization is based on a lowest-order face-based scheme supporting polytopal meshes, namely discrete velocities are attached to the mesh faces and cells, whereas discrete pressures are attached to the mesh cells. This face-based scheme can be embedded into the framework of hybrid mixed mimetic schemes and gradient schemes, and has close links to the lowest-order version of hybrid high-order methods devised for the steady incompressible Navier–Stokes equations. The AC time-stepping uncouples at each time step the velocity update from the pressure update. The performances of this approach are compared against those of the more traditional monolithic approach which maintains the velocity-pressure coupling at each time step. We consider both first-order and second-order time schemes and either an implicit or an explicit treatment of the nonlinear convection term. We investigate numerically the CFL stability restriction resulting from an explicit treatment, both on Cartesian and polytopal meshes. Finally, numerical tests on large 3D polytopal meshes highlight the efficiency of the AC approach and the benefits of using second-order schemes whenever accurate discrete solutions are to be attained.

Keywords: incompressible Navier–Stokes, artificial compressibility, polytopal meshes, lowest-order hybrid schemes

MSC (2010): 65M12, 65M22, 76D05, 76M10

1 Introduction

The goal of this work is to study the accuracy and efficiency of artificial compressibility techniques for the time discretization of the incompressible Navier–Stokes equations. These equations are encountered in a wide range of industrial applications, ranging from aeronautics to the simulations of flows in micro-fractures, to cite two salient examples. In this work, the space discretization is realized by means of lowest-order face-based schemes supporting classical (simplicial or Cartesian) as well as general (polytopal) meshes. The support of polytopal meshes is motivated by the applications, especially in the industrial context. Indeed, such meshes often alleviate substantially the burden of mesh generation resulting from the complexity of the geometry (as, e.g., the shape of industrial equipments) and/or the heterogeneity of the physical properties (requiring local mesh refinements producing hanging nodes).

The unsteady incompressible Navier–Stokes equations read as follows:

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + (\underline{u} \cdot \underline{\nabla})\underline{u} + \underline{\nabla}\, p = \underline{f}\,, \\ \underline{\nabla} \cdot \underline{u} = 0\,, \end{cases} \qquad \text{in } \Omega \times (0,T)\,, \tag{1}$$

[*]EDF R&D, 6 Quai Watier, 78400 Chatou, France

[†]CERMICS, Ecole des Ponts, 77455 Marne-la-Vallée 2, France, and Inria, 2 rue Simone Iff, 75589 Paris, France

together with suitable boundary and initial conditions on the velocity. Here, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is an open, polytopal, bounded, Lipschitz domain and $T > 0$ denotes the observation time. The unknowns are the velocity $\underline{u}$ and the pressure p (by convention, vector-valued quantities are underlined). The problem data are the viscosity $\nu > 0$, the mass density is normalized to unity, and the body force is denoted by $\underline{f}$. For simplicity, we consider smooth solutions to (1), and we enforce Dirichlet boundary conditions over the whole boundary $\partial\Omega$ at all times. Recall that the first equation in (1) expresses the momentum balance in the flow, and the second equation the mass conservation.

Let us briefly describe the time-marching schemes considered in this work without introducing any space discretization yet. These schemes are either of monolithic form, thereby requiring to solve a saddle point problem at each time step, or they uncouple the velocity and the pressure at each time step by means of an artificial compressibility technique. Let $\Delta t > 0$ be the time step (taken to be constant for simplicity) and let $(\underline{u}^n, p^n)$ denote the approximate solution at the discrete time node $t^n \coloneqq n\Delta t$ for all $n \geq 1$. The monolithic approach is the traditional time-marching scheme for the incompressible Navier–Stokes equations and it reads as follows: For all $n \geq 1$, given $\underline{u}^{n-1}$ from the initial condition or the previous time node, find $(\underline{u}^n, p^n)$ by solving the saddle point problem in Ω,

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + (\underline{u}^n \cdot \underline{\nabla})\underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n \coloneqq \underline{f}(t^n)\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0\,. \end{cases} \tag{2}$$

The monolithic approach is well-known and is considered here as the reference time-stepping scheme. The only approximation introduced in (2) with respect to (1) is the discretization of the velocity time-derivative (here, by means of the implicit Euler scheme to fix the ideas). In contrast, the artificial compressibility (AC) approach introduces an additional approximation at each time step in that the velocity is first updated and the pressure is then corrected. In its first-order form, the scheme reads as follows: For all $n \geq 1$, given $(\underline{u}^{n-1}, p^{n-1})$ from the initial condition or the previous time node, find $(\underline{u}^n, p^n)$ by solving a parabolic-like problem on the velocity and then updating the pressure:

$$\frac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu(\underline{\Delta}\, \underline{u}^n + \eta \underline{\nabla}\, \underline{\nabla} \cdot \underline{u}^n) + (\underline{u}^n \cdot \underline{\nabla})\underline{u}^n = \underline{f}^n - \underline{\nabla}\, p^{n-1}\,, \tag{3a}$$

$$p^n = p^{n-1} - \nu\eta \underline{\nabla} \cdot \underline{u}^n\,, \tag{3b}$$

where $\eta > 0$ is a user-defined adimensional parameter. The AC method was introduced in the late sixties [23, 56] in the Western literature, although the seminal ideas can be traced independently in the Russian literature as well [57, 49, 58]. For a recent analysis of the method, the reader is referred to [45]. Notice the appearance of the additional grad-div term in (3a), whereas (3b) shows that the time discrete velocity field $\underline{u}^n$ is no longer divergence-free. Notice also that an approximation to the initial pressure is needed. Furthermore, we observe that higher-order versions of the AC scheme are available; see [45, 44].

The computational study performed in this work considers both the monolithic and the AC approaches, either in the above form which is first-order accurate in time, or in second-order form (as described below). Moreover, we examine either the implicit treatment of the convection term (as above) leading to a nonlinear problem to be solved at each time step, or a (semi-)explicit treatment leading to a linear problem to be solved at each time step. The natural choice when using the AC scheme is to resort to an explicit treatment since the goal is to reduce as much as possible the computational cost per time step. However, for completeness, we also consider the implicit treatment of the convection term for the first-order AC schemes, but not for the second-order ones. Indeed, in this latter case, there are two nonlinear problems to be solved at each time step, which is deemed to be too expensive.

Concerning the space discretization, we focus on schemes supporting polytopal meshes. Numerous possibilities are available from the literature. Concerning high-order schemes, we mention (without being exhaustive) discontinuous Galerkin (dG) methods [7, 24, 46], hybridizable dG (HDG) methods [25, 53, 47, 50], hybrid high-order (HHO) methods [1, 29, 30, 17], weak Galerkin (WG) methods [52], virtual element methods (VEM) [9, 10, 42], and nonconforming VEM [20]. As efficient as the above high-order methods can be, it is often preferred in an industrial context to use lowest-order schemes owing to their relative ease of implementation and the prominence of legacy codes based on these techniques. Moreover, since the maximal order of accuracy in time considered here is two, it is reasonable to focus on lowest-order space discretization schemes. Examples of such schemes for the (Navier–)Stokes equations include discrete duality finite volume (DDFV) schemes [26, 48, 18, 43], mixed finite volume (MFV) schemes [32], mimetic finite difference (MFD) schemes [8], compatible discrete operator (CDO) schemes [15, 12], and gradient schemes [40, 34, 37]. Unifying frameworks bridging a large class of lowest-order schemes exist, such as the hybrid mixed mimetic (HMM) framework from [35] and the setting of gradient schemes [35]. In this work, we focus more specifically on CDO schemes. This is a mild restriction, and we expect that most of our conclusions can be carried out to other lowest-order schemes owing to the above-mentioned unifying frameworks. Among the various CDO schemes, we focus on face-based CDO (CDO-Fb) schemes in which the degrees of freedom (DoFs) for the velocity are vector-valued and attached to the mesh faces and cells, whereas the pressure DoFs are scalar-valued and attached to the mesh cells only. We refer the reader to [14] for the analysis of vertex-based and cell-based CDO schemes for elliptic PDEs and to [21, 22] for edge-based CDO schemes applied to circulations (i.e., differential forms of order one). The CDO schemes devised in [15] are different since they introduce the vorticity as an additional unknown. As discussed in [12, Sect. 8.3], CDO-Fb schemes are derived for cell-based schemes by a hybridization procedure of the flux unknown considered in cell-based schemes. CDO-Fb schemes can be bridged to lowest-order HHO, HFV, HMM, and gradient schemes, and have been evaluated numerically to approximate the steady incompressible Navier–Stokes equations in [16].

This paper is organized as follows. The CDO-Fb scheme for the space discretization is presented in Section 2. The fully discrete schemes are introduced in Section 3 using either the monolithic or the AC approaches and either an implicit or an explicit treatment of the convective term. Numerical results are discussed in Section 4 on two- and three-dimensional test cases. Finally, conclusions are drawn in Section 5.

2 Space discretization by the CDO-Fb scheme

In this section, we recall the CDO-Fb scheme for the space discretization of the steady incompressible Navier–Stokes equations introduced in [16]. In weak form, and considering homogeneous Dirichlet conditions for simplicity (and fixing the pressure mean value to zero), one seeks $(\underline{u}, p) \in \underline{H}^1_0(\Omega) \times L^2_0(\Omega)$ (standard notation for Lebesgue and Sobolev spaces is employed) such that

$$\begin{cases} \nu a(\underline{u}, \underline{v}) + t(\underline{u}; \underline{u}, \underline{v}) + b(\underline{v}, p) = l(\underline{v}) \,, & \forall \underline{v} \in \underline{H}^1_0(\Omega) \,, \\ b(\underline{u}, q) = 0 \,, & \forall q \in L^2_0(\Omega) \,, \end{cases} \tag{4}$$

with the bilinear and trilinear forms

$$a(\underline{u}, \underline{v}) \coloneqq \int_\Omega \underline{\underline{\nabla}}\, \underline{u} : \underline{\underline{\nabla}}\, \underline{v} \,, \qquad b(\underline{v}, p) \coloneqq - \int_\Omega (\underline{\nabla} \cdot \underline{v}) p \,, \qquad t(\underline{w}; \underline{u}, \underline{v}) \coloneqq \int_\Omega \big((\underline{w} \cdot \underline{\nabla})\underline{u}\big) \cdot \underline{v} \,, \tag{5}$$

and the linear form $l(\underline{v}) \coloneqq \int_\Omega \underline{f} \cdot \underline{v}$.

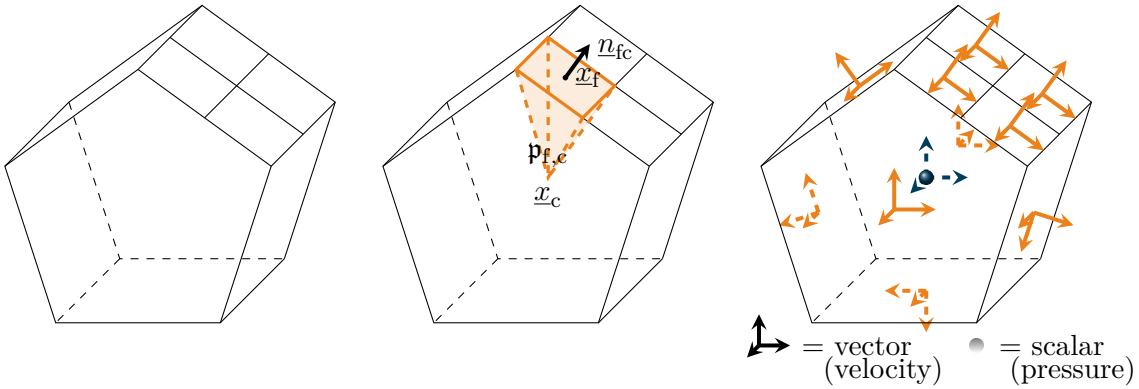

Figure 1: Mesh, notation and DoFs. Left: Example of a polyhedral cell with a hanging node. Center: The barycenter $\underline{x}_c$ of the cell c and a face $f \in F_c$ with its barycenter, $\underline{x}_f$, its normal $\underline{n}_{fc}$ and its sub-pyramid $\mathfrak{p}_{f,c}$. Right: Full set of velocity and pressure DoFs for the considered cell (dashed arrows are used for velocity DoFs on hidden faces).

2.1 Mesh entities and degrees of freedom

The mesh discretizing Ω is a finite collection $C \coloneqq \{c\}$ of nonempty, disjoint, open, polytopal subsets of $\mathbb{R}^d$, usually referred to as cells. The mesh faces are assumed to be planar and are gathered in the set $F \coloneqq \{f\}$ which can be subdivided in the two disjoint sets $F^b \coloneqq \{f \,|\, f \subset \partial\Omega\}$ collecting the mesh boundary faces and $F^i \coloneqq F \backslash F^b$ collecting the mesh interfaces. One associates with each mesh face f a normal vector $\underline{n}_f$ such that if $f \in F^b$, $\underline{n}_f$ points outward Ω and, if $f \in F^i$, the direction is chosen arbitrarily and fixed once and for all. For a generic mesh entity $z = c$ or $z = f$, $\underline{x}_z$ denotes its barycenter, $|z|$ its measure and $h_z \coloneqq \max_{\underline{x}_1,\underline{x}_2 \in z} |\underline{x}_1 - \underline{x}_2|$ its diameter. Moreover, $h \coloneqq \max_{c\in C} h_c$ is called the size of the mesh. We consider the shape-regularity setting of [28] for polytopal mesh families, and we additionally require that every mesh entity $z = c$ or $z = f$ is star-shaped with respect to its barycenter $\underline{x}_z$. The faces composing the boundary of a cell $c \in C$ are collected in the set $F_c \coloneqq \{f \in F \,|\, f \subset \partial c\}$. For each face $f \in F_c$, a unit normal vector pointing outward c is denoted by $\underline{n}_{fc} = \pm\underline{n}_f$, depending on the orientation chosen for $\underline{n}_f$. We are going to make use of a subdivision of the cell c as $\mathfrak{P}_c \coloneqq \{\mathfrak{p}_{f,c}\}_{f\in F_c}$, where the subsets $\mathfrak{p}_{f,c}$ are the nonempty, disjoint subpyramids (or subtriangles if $d = 2$) obtained by considering the cell barycenter $\underline{x}_c$ as apex, and a face $f \in F_c$ as basis.

For a generic mesh entity $z = c$ or $z = f$, $\mathbb{P}^q(z)$ (resp., $\underline{\mathbb{P}}^q(z)$ and $\underline{\underline{\mathbb{P}}}^q(z)$) is composed of the restriction to z of the scalar-valued (resp., $\mathbb{R}^d$-valued and $\mathbb{R}^{d\times d}$-valued) polynomials of degree at most q. Moreover, for a collection Z of mesh entities, $\mathbb{P}^q(Z)$, $\underline{\mathbb{P}}^q(Z)$, and $\underline{\underline{\mathbb{P}}}^q(Z)$ refer to piecewise-polynomial functions; for instance, $\mathbb{P}^1(F_c) \coloneqq \bigtimes_{f\in F_c} \mathbb{P}^1(f)$. Notice also that for $z = c$ or $z = f$, $\mathbb{P}^0(z) \equiv \mathbb{R}$.

In the CDO-Fb framework, the velocity is hybrid, meaning that it has cell- and face-based DoFs. Hence, the global velocity space is

$$\widehat{\underline{\mathcal{U}}}_h \coloneqq \bigtimes_{f\in F} \underline{\mathbb{P}}^0(f) \times \bigtimes_{c\in C} \underline{\mathbb{P}}^0(c)\,. \tag{6}$$

A generic element of $\widehat{\underline{\mathcal{U}}}_h$ is denoted by $\widehat{\underline{v}}_h \coloneqq ((\underline{v}_f)_{f\in F}, (\underline{v}_c)_{c\in C})$. Notice that the velocity DoFs at the interfaces are uniquely defined. Moreover, the velocity DoFs associated with a generic cell $c \in C$ are denoted by

$$\widehat{\underline{v}}_c \coloneqq ((\underline{v}_f)_{f\in F_c}, \underline{v}_c) \in \widehat{\underline{\mathcal{U}}}_c \coloneqq \bigtimes_{f\in F_c} \underline{\mathbb{P}}^0(f) \times \underline{\mathbb{P}}^0(c)\,. \tag{7}$$

The pressure has only cell-based DoFs, so that the global pressure space is

$$\mathcal{P}_{\mathrm{h}} \coloneqq \bigtimes_{\mathrm{c}\in\mathrm{C}} \mathcal{P}_{\mathrm{c}} \ni p_{\mathrm{h}} \coloneqq (p_{\mathrm{c}})_{\mathrm{c}\in\mathrm{C}}\,, \qquad \mathcal{P}_{\mathrm{c}} \coloneqq \mathbb{P}^0(\mathrm{c})\,. \tag{8}$$

In order to account for the velocity boundary conditions and the constraint on the pressure average, we consider the subspaces

$$\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \coloneqq \{\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \,|\, \underline{v}_{\mathrm{f}} = \underline{0}\ \forall \mathrm{f}\in\mathrm{F}^b\}, \qquad \mathcal{P}_{\mathrm{h},*} \coloneqq \{p_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h}} \,|\, \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, p_{\mathrm{c}} = 0\}. \tag{9}$$

Finally, for $\mathrm{z} = \mathrm{c}$ or $\mathrm{z} = \mathrm{f}$, π_{z} (resp., $\underline{\pi}_{\mathrm{z}}$) denotes the L^2-orthogonal projection onto $\mathbb{P}^0(\mathrm{z})$ (resp., $\underline{\mathbb{P}}^0(\mathrm{z})$). The L^2-orthogonal projection onto $\mathcal{P}_{\mathrm{h}}$ is defined cellwise so that $\pi_{\mathrm{h}}(q) \coloneqq (\pi_{\mathrm{c}}(q|_{\mathrm{c}}))_{\mathrm{c}\in\mathrm{C}}$ for all $q \in L^1(\Omega)$, whereas the projection onto the hybrid discrete space $\widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$ is defined such that $\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v}) \coloneqq \big((\underline{\pi}_{\mathrm{f}}(\underline{v}|_{\mathrm{f}}))_{\mathrm{f}\in\mathrm{F}}, (\underline{\pi}_{\mathrm{c}}(\underline{v}))_{\mathrm{c}\in\mathrm{C}}\big)$ for all $\underline{v} \in \underline{H}^s(\Omega)$, $s > \frac{1}{2}$. Similarly, for the local hybrid space $\widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$, we set $\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v}) \coloneqq \big((\underline{\pi}_{\mathrm{f}}(\underline{v}|_{\mathrm{f}}))_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}, \underline{\pi}_{\mathrm{c}}(\underline{v})\big)$ for all $\underline{v} \in \underline{H}^s(\mathrm{c})$.

2.2 Discrete diffusion-like bilinear form

Consider a cell $\mathrm{c}\in\mathrm{C}$ and the velocity DoFs $\widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$ associated with c. We define locally in the mesh cell c a velocity gradient reconstruction operator, $\underline{\underline{\mathrm{G}}}_{\mathrm{c}} : \widehat{\underline{\mathcal{U}}}_{\mathrm{c}} \to \underline{\underline{\mathbb{P}}}^0(\mathfrak{P}_{\mathrm{c}})$, i.e., for all $\widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$, $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})$ is a piecewise constant tensor-valued field on the subpyramids collected in $\mathfrak{P}_{\mathrm{c}}$. In the context of CDO schemes, this type of operator was introduced in [12]; see also [13]. It is composed of a consistent part, $\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}$, plus a stabilization part, $\underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}$, so that

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) \coloneqq \underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) + \underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\,, \tag{10}$$

where $\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) \in \underline{\underline{\mathbb{P}}}^0(\mathrm{c}) \subset \underline{\underline{\mathbb{P}}}^0(\mathfrak{P}_{\mathrm{c}})$ and $\underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) \in \underline{\underline{\mathbb{P}}}^0(\mathfrak{P}_{\mathrm{c}})$ are defined as follows:

$$\begin{aligned} \underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) &\coloneqq \frac{1}{|\mathrm{c}|}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}}\,, \\ \underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})|_{\mathfrak{p}_{\mathrm{f,c}}} &\coloneqq \theta\frac{|\mathrm{f}|}{|\mathfrak{p}_{\mathrm{f,c}}|}\Big((\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) - \underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}})\Big) \otimes \underline{n}_{\mathrm{fc}}\,, \qquad \forall \mathrm{f}\in\mathrm{F}_{\mathrm{c}}\,. \end{aligned} \tag{11}$$

The positive stabilization parameter θ is user-defined (with the only requirement of being positive): choosing $\theta \coloneqq 1$ recovers the generalized Crouzeix–Raviart scheme from [31], whereas $\theta \coloneqq \frac{1}{\sqrt{d}}$ leads to the HFV scheme from [38].

The global version of the gradient reconstruction operator, $\underline{\underline{\mathrm{G}}}_{\mathrm{h}} : \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \to \bigtimes_{\mathrm{c}\in\mathrm{C}} \underline{\underline{\mathbb{P}}}^0(\mathfrak{P}_{\mathrm{c}})$, is defined cellwise so that, for any $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$ and any $\mathrm{c}\in\mathrm{C}$, we have $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})|_{\mathrm{c}} \coloneqq \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})$. With the above operators in hand, the discrete diffusion-like bilinear form $a_h : \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \times \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \to \mathbb{R}$ is defined as follows:

$$a_h(\widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \int_\Omega \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}) : \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) = \sum_{\mathrm{c}\in\mathrm{C}} \int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}}) : \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\,. \tag{12}$$

The above-defined gradient reconstruction operators enjoy several important properties. First, $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$ is consistent for affine functions, meaning that

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v})) = \underline{\underline{\nabla}}\underline{v}\,, \qquad \forall \underline{v} \in \underline{\mathbb{P}}^1(\mathrm{c})\,. \tag{13}$$

(More precisely, we have $\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v})) = \underline{\underline{\nabla}}\underline{v}$ and $\underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v})) = \underline{0}$.) Moreover, the consistent and stabilization parts of the reconstructed gradient are $\underline{\underline{L}}^2(\mathrm{c})$-orthogonal,

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) : \underline{\underline{\mathrm{G}}}^{\mathrm{s}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) = 0\,, \qquad \forall \widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}\,. \tag{14}$$

Finally, and most importantly, the following stability and boundedness properties hold true: There is $\delta > 0$ such that for all $\mathrm{c} \in \mathrm{C}$ and all $\widehat{\underline{v}}_\mathrm{c} \in \widehat{\underline{\mathcal{U}}}_\mathrm{c}$,

$$\delta \|\widehat{\underline{v}}_\mathrm{c}\|_{1,\mathrm{c}}^2 \le \big\|\underline{\underline{\mathrm{G}}}_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c})\big\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 \le \delta^{-1} \|\widehat{\underline{v}}_\mathrm{c}\|_{1,\mathrm{c}}^2 , \tag{15}$$

with the discrete H^1-like seminorm defined on $\widehat{\underline{\mathcal{U}}}_\mathrm{c}$ such that $\|\widehat{\underline{v}}_\mathrm{c}\|_{1,\mathrm{c}}^2 \coloneqq \sum_{\mathrm{f}\in\mathrm{F}_\mathrm{c}} h_\mathrm{c}^{-1} |\mathrm{f}| \, |\underline{v}_\mathrm{f} - \underline{v}_\mathrm{c}|^2$. The lower bound (15) is the main reason for introducing the stabilization part of the reconstructed gradient. Notice that $\|\widehat{\underline{v}}_\mathrm{h}\|_{1,\mathrm{h}}^2 \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} \|\widehat{\underline{v}}_\mathrm{c}\|_{1,\mathrm{c}}^2$ defines a norm on $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$.

2.3 Discrete velocity-pressure coupling

The discrete velocity-pressure coupling hinges on a discrete divergence operator $\mathrm{D}_\mathrm{h} \colon \widehat{\underline{\mathcal{U}}}_\mathrm{h} \to \mathcal{P}_\mathrm{h}$ that is defined cellwise. For all $\mathrm{c} \in \mathrm{C}$, the local discrete divergence operator $\mathrm{D}_\mathrm{c} \colon \widehat{\underline{\mathcal{U}}}_\mathrm{c} \to \mathcal{P}_\mathrm{c}$ is such that for all $\widehat{\underline{v}}_\mathrm{c} \in \widehat{\underline{\mathcal{U}}}_\mathrm{c}$,

$$\mathrm{D}_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c}) \coloneqq \mathrm{tr}\left(\underline{\underline{\mathrm{G}}}_\mathrm{c}^0(\widehat{\underline{v}}_\mathrm{c})\right) = \frac{1}{|\mathrm{c}|} \sum_{\mathrm{f}\in\mathrm{F}_\mathrm{c}} |\mathrm{f}| \, (\underline{v}_\mathrm{f} - \underline{v}_\mathrm{c}) \cdot \underline{n}_\mathrm{fc} . \tag{16}$$

The global operator is then defined by setting $\mathrm{D}_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h})|_\mathrm{c} = \mathrm{D}_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c})$ for all $\widehat{\underline{v}}_\mathrm{h} \in \widehat{\underline{\mathcal{U}}}_\mathrm{h}$ and all $\mathrm{c} \in \mathrm{C}$. Notice that $\mathrm{D}_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}) \in \mathcal{P}_{\mathrm{h},*}$ for all $\underline{v}_\mathrm{h} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$.

With the above divergence operator in hand, the discrete bilinear form $b_h : \widehat{\underline{\mathcal{U}}}_\mathrm{h} \times \mathcal{P}_\mathrm{h} \to \mathbb{R}$ handling the velocity-pressure coupling is defined as follows:

$$b_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}, q_\mathrm{h}) \coloneqq -\int_\Omega \mathrm{D}_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}) q_\mathrm{h} = -\sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}| \, \mathrm{D}_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c}) q_\mathrm{c} . \tag{17}$$

The same discrete bilinear form is found in HMM schemes [34, 33] and the lowest-order HHO scheme [1, 29]. A crucial property is the following discrete inf-sup condition: There is $\beta > 0$, only depending on the mesh shape-regularity, such that

$$\inf_{q_\mathrm{h}\in\mathcal{P}_{\mathrm{h},*}} \sup_{\widehat{\underline{v}}_\mathrm{h}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}} \frac{|b_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}, q_\mathrm{h})|}{\|q_\mathrm{h}\|_\mathrm{h} \, \|\widehat{\underline{v}}_\mathrm{h}\|_{1,\mathrm{h}}} \ge \beta , \tag{18}$$

where $\|\widehat{\underline{v}}_\mathrm{h}\|_{1,\mathrm{h}}$ is defined above whereas $\|q_\mathrm{h}\|_\mathrm{h}^2 \coloneqq \|q_\mathrm{h}\|_{L^2(\Omega)}^2 = \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}| \, |q_\mathrm{c}|^2$ for all $q_\mathrm{h} \in \mathcal{P}_\mathrm{h}$. Another important property of the discrete divergence operator is its commuting property with the L^2-orthogonal projection in the sense that $\mathrm{D}_\mathrm{c}(\widehat{\underline{\pi}}_\mathrm{c}(\underline{v})) = \pi_\mathrm{c}(\underline{\nabla} \cdot \underline{v})$ for all $\mathrm{c} \in \mathrm{C}$ and all $\underline{v} \in \underline{H}^1(\mathrm{c})$.

2.4 Discrete convection operator

Let us finally devise a discrete CDO-Fb counterpart of the trilinear form t defined in (5). To this purpose, we define the discrete trilinear form $t_h : \widehat{\underline{\mathcal{U}}}_\mathrm{h} \times \widehat{\underline{\mathcal{U}}}_\mathrm{h} \times \widehat{\underline{\mathcal{U}}}_\mathrm{h} \to \mathbb{R}$ such that

$$t_\mathrm{h}(\widehat{\underline{w}}_\mathrm{h}; \widehat{\underline{u}}_\mathrm{h}, \widehat{\underline{v}}_\mathrm{h}) \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} t_\mathrm{c}(\widehat{\underline{w}}_\mathrm{c}; \widehat{\underline{u}}_\mathrm{c}, \widehat{\underline{v}}_\mathrm{c}) + \sum_{\mathrm{f}\in\mathrm{F}^b} t_\mathrm{f}(\underline{w}_\mathrm{f}; \underline{u}_\mathrm{f}, \underline{v}_\mathrm{f}) , \tag{19}$$

with

$$\begin{aligned} t_\mathrm{c}(\widehat{\underline{w}}_\mathrm{c}; \widehat{\underline{u}}_\mathrm{c}, \widehat{\underline{v}}_\mathrm{c}) &\coloneqq \frac{1}{2} \sum_{\mathrm{f}\in\mathrm{F}_\mathrm{c}} |\mathrm{f}| \, (\underline{w}_\mathrm{f} \cdot \underline{n}_\mathrm{fc})(\underline{u}_\mathrm{f} - \underline{u}_\mathrm{c}) \cdot (\underline{v}_\mathrm{f} + \underline{v}_\mathrm{c}) , \\ t_\mathrm{f}(\underline{w}_\mathrm{f}; \underline{u}_\mathrm{f}, \underline{v}_\mathrm{f}) &\coloneqq |\mathrm{f}| \, (\underline{w}_\mathrm{f} \cdot \underline{n}_\mathrm{fc})^- \underline{u}_\mathrm{f} \cdot \underline{v}_\mathrm{f} , \end{aligned} \tag{20}$$

where $(x)^- \coloneqq \frac{1}{2}(|x| - x)$ denotes the negative part of any real number $x \in \mathbb{R}$. The trilinear form t_f is related to the weak enforcement of inflow boundary conditions and vanishes whenever

at least one of the arguments is in $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$. The discrete trilinear form t_h is inspired from the lowest-order HHO method for scalar advection-diffusion equations introduced in [27]; see also [17, Remark 9] for the Navier–Stokes equations.

The first important property of the discrete trilinear form t_h is its positivity and skew-symmetry. Indeed, assume that $\widehat{\underline{w}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$ is discretely divergence-free, i.e., $\mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}}) = 0$ for all $\mathrm{c} \in \mathrm{C}$. Then, one readily verifies that

$$t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{u}}_{\mathrm{h}}) \geq 0\,, \qquad \forall \widehat{\underline{u}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}\,. \tag{21}$$

Moreover, if additionally the normal component of $\widehat{\underline{w}}_{\mathrm{h}}$ vanishes at the boundary, namely $\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}} = 0$ for all $\mathrm{f} \in \mathrm{F}^b$, then $t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}; \cdot, \cdot)$ is skew-symmetric, i.e.

$$t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{u}}_{\mathrm{h}}) = 0\,, \qquad \forall \widehat{\underline{u}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}\,. \tag{22}$$

Property (21) is crucial in the context of the Navier–Stokes equations since it is instrumental to establish the dissipativity of the discrete problem. Notice also that (22) is reminiscent of the so-called Temam's trick on the discrete trilinear form. A further relevant property of the discrete trilinear form t_h is its limit-conformity. This notion is described, e.g., in [37], and is verified for the above CDO-Fb setting in [51, Remark 2.50]; see also [30, Prop. 6].

Remark 1 (Upwinding). An upwinding stabilization can be considered by adding to (19) the discrete trilinear form $t_{\mathrm{h}}^{\mathrm{u}}(\widehat{\underline{w}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \frac{1}{2} \sum_{\mathrm{f} \in \mathrm{F}^i} \sum_{\mathrm{c} \in \mathrm{C}_{\mathrm{f}}} |\mathrm{f}|\, |\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{fc}}|\, (\underline{u}_{\mathrm{f}} - \underline{u}_{\mathrm{c}}) \cdot (\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}})$. This option is not considered further in the present work.

3 Time discretization by monolithic and artificial compressibility schemes

We present in this section the fully discrete problems obtained by using either a monolithic or an AC scheme for the time discretization and by considering either an implicit or an explicit treatment of the convection operator. Moreover, we consider both first-order and second-order time schemes.

For a hybrid velocity field $\widehat{\underline{v}}_{\mathrm{h}} \coloneqq ((\underline{v}_{\mathrm{f}})_{\mathrm{f} \in \mathrm{F}}, (\underline{v}_{\mathrm{c}})_{\mathrm{c} \in \mathrm{C}}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$, we denote its cell-based components by $\underline{v}_{\mathrm{C}} \coloneqq (\underline{v}_{\mathrm{c}})_{\mathrm{c} \in \mathrm{C}}$. Then, to discretize the time-derivative of the velocity, we only use the cell-based components and consider the mass bilinear form such that

$$m(\underline{u}_{\mathrm{C}}, \underline{v}_{\mathrm{C}}) \coloneqq \sum_{\mathrm{c} \in \mathrm{C}} |\mathrm{c}|\, \underline{u}_{\mathrm{c}} \cdot \underline{v}_{\mathrm{c}}\,. \tag{23}$$

We also use the cell-based components to discretize the source term and we set $l^n(\underline{v}_{\mathrm{C}}) \coloneqq \sum_{\mathrm{c} \in \mathrm{C}} \int_{\mathrm{c}} \underline{f}^n \cdot \underline{v}_{\mathrm{c}}$ for all $n \geq 1$. For any hybrid velocity field $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, we define its discrete kinetic energy as

$$\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \frac{1}{2} \|\underline{v}_{\mathrm{C}}\|^2_{\underline{L}^2(\Omega)} = \frac{1}{2} \sum_{\mathrm{c} \in \mathrm{C}} |\mathrm{c}|\, |\underline{v}_{\mathrm{c}}|_2^2\,. \tag{24}$$

3.1 First-order schemes

3.1.1 Monolithic scheme

Let us first consider an implicit treatment of the convection term which is dealt with by means of a Picard algorithm. Then, the first-order monolithic time-stepping scheme reads as follows. For all $n = 1, \ldots, N$, iterate on $k \geq 1$ until convergence: Find $(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, p_{\mathrm{h}}^{n,k}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n,k} - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n,k}) = l^n(\underline{v}_{\mathrm{C}})\,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, q_{\mathrm{h}}) = 0\,. \end{cases} \tag{25}$$

for all $\widehat{\underline{v}}_{\rm h} \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0}$ and all $q_{\rm h} \in \mathcal{P}_{{\rm h},*}$. Here, $\widehat{\underline{u}}_{\rm h}^{n-1}$ denotes the solution given by the Picard algorithm at the time step $n-1$. The time-stepping is initialized with the initial condition $\widehat{\underline{u}}_{\rm h}^{0} \coloneqq \widehat{\underline{\pi}}_{\rm h}(\underline{u}_0)$. Moreover, at each time step, the Picard iterations have to be initialized: a suitable choice is to take the solution at the previous time step, i.e., $\widehat{\underline{u}}_{\rm h}^{n,0} \coloneqq \widehat{\underline{u}}_{\rm h}^{n-1}$. Taking $\widehat{\underline{v}}_{\rm h} = \widehat{\underline{u}}_{\rm h}^{n,k}$ in (25) and using simple arithmetic manipulations together with $t_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k-1}; \widehat{\underline{u}}_{\rm h}^{n,k}, \widehat{\underline{u}}_{\rm h}^{n,k}) = 0$ owing to (22), we obtain the following discrete kinetic energy balance:

$$\mathcal{E}_{\rm kin,h}(\widehat{\underline{u}}_{\rm h}^{n,k}) - \mathcal{E}_{\rm kin,h}(\widehat{\underline{u}}_{\rm h}^{n-1}) + \mathcal{E}_{\rm kin,h}(\widehat{\underline{u}}_{\rm h}^{n,k} - \widehat{\underline{u}}_{\rm h}^{n-1}) + \nu\Delta t \|\underline{\underline{\mathrm{G}}}_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k})\|_{\underline{\underline{L}}^2(\Omega)}^2 = \Delta t\, l^n(\underline{u}_{\rm C}^{n,k})\,. \tag{26}$$

Notice that the third and fourth terms on the left-hand side are non-negative, i.e., they only contribute to energy dissipation. Notice also that (26) remains valid as $k \to \infty$.

With an explicit treatment of the convection term, the first-order monolithic time-stepping scheme reads as follows: For all $n = 1, \dots, N$, find $(\underline{u}_{\rm h}^n, p_{\rm h}^n) \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0} \times \mathcal{P}_{{\rm h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\rm C}^n - \underline{u}_{\rm C}^{n-1}, \underline{v}_{\rm C}) + \nu a_{\rm h}(\widehat{\underline{u}}_{\rm h}^n, \widehat{\underline{v}}_{\rm h}) + b_{\rm h}(\widehat{\underline{v}}_{\rm h}, p_{\rm h}^n) = l^n(\underline{v}_{\rm C}) - t_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n-1}; \widehat{\underline{u}}_{\rm h}^{n-1}, \widehat{\underline{v}}_{\rm h})\,, \\ b_{\rm h}(\widehat{\underline{u}}_{\rm h}^n, q_{\rm h}) = 0\,, \end{cases} \tag{27}$$

for all $\widehat{\underline{v}}_{\rm h} \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0}$ and all $q_{\rm h} \in \mathcal{P}_{{\rm h},*}$. Unfortunately, a dissipative discrete kinetic energy balance cannot be derived for (27) because the term $t_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n-1}; \widehat{\underline{u}}_{\rm h}^{n-1}, \widehat{\underline{u}}_{\rm h}^{n})$ cannot be given an a priori sign. Thus, the scheme (27) is subject to a first-order CFL restriction on the time step for its stability. This condition will be investigated numerically in Section 4.

3.1.2 AC scheme

Let us first consider an implicit treatment of the convection term. As in the monolithic approach, we consider Picard iterations to deal with the resulting nonlinearity. Then, the first-order AC time-stepping scheme reads as follows. For all $n = 1, \dots, N$, iterate on $k \geq 1$ until convergence: Find $\underline{u}_{\rm h}^{n,k} \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0}$ such that for all $\widehat{\underline{v}}_{\rm h} \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0}$,

$$\begin{aligned} \frac{1}{\Delta t} m(\underline{u}_{\rm C}^{n,k} - \underline{u}_{\rm C}^{n-1}, \underline{v}_{\rm C}) + \nu(a_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k}, \widehat{\underline{v}}_{\rm h}) + \eta d_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k}, \widehat{\underline{v}}_{\rm h})) + t_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k-1}; \widehat{\underline{u}}_{\rm h}^{n,k}, \widehat{\underline{v}}_{\rm h}) \\ = l^n(\underline{v}_{\rm C}) - b_{\rm h}(\widehat{\underline{v}}_{\rm h}, p_{\rm h}^{n,k-1})\,, \end{aligned} \tag{28}$$

and then set

$$p_{\rm h}^{n,k} = p_{\rm h}^{n,k-1} - \nu\eta \mathrm{D}_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n,k})\,, \tag{29}$$

recalling that $\eta > 0$ is a user-defined parameter (choices are discussed in Section 4). In (28), we introduced the discrete div-div bilinear form such that

$$d_{\rm h}(\widehat{\underline{u}}_{\rm h}, \widehat{\underline{v}}_{\rm h}) \coloneqq \int_\Omega \mathrm{D}_{\rm h}(\widehat{\underline{u}}_{\rm h})\mathrm{D}_{\rm h}(\widehat{\underline{v}}_{\rm h}) = \sum_{\rm c\in C} |c|\, \mathrm{D}_{\rm c}(\widehat{\underline{u}}_{\rm c})\mathrm{D}_{\rm c}(\widehat{\underline{v}}_{\rm c})\,. \tag{30}$$

Notice that in (28)-(29), we took advantage of the Picard iterations to update progressively the pressure. Letting $k \to \infty$, we observe that the velocity and pressure updates stemming from the monolithic and the AC schemes are the same. Differently from the monolithic coupling, the initialization of the Picard iteration requires to specify the velocity and the pressure, which are here taken from the previous time step or the initial condition.

With an explicit treatment of the convection term, the first-order AC scheme reads as follows: For all $n = 1, \dots, N$, find $(\underline{u}_{\rm h}^n, p_{\rm h}^n) \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0} \times \mathcal{P}_{{\rm h},*}$ such that for all $\widehat{\underline{v}}_{\rm h} \in \widehat{\underline{\mathcal{U}}}_{{\rm h},0}$,

$$\begin{aligned} \frac{1}{\Delta t} m(\underline{u}_{\rm C}^{n} - \underline{u}_{\rm C}^{n-1}, \underline{v}_{\rm C}) + \nu(a_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n}, \widehat{\underline{v}}_{\rm h}) + \eta d_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n}, \widehat{\underline{v}}_{\rm h})) \\ = l^n(\underline{v}_{\rm C}) - b_{\rm h}(\widehat{\underline{v}}_{\rm h}, p_{\rm h}^{n-1}) - t_{\rm h}(\widehat{\underline{u}}_{\rm h}^{n-1}; \widehat{\underline{u}}_{\rm h}^{n-1}, \widehat{\underline{v}}_{\rm h})\,, \end{aligned} \tag{31}$$

and then set

$$p_{\mathrm{h}}^{n} = p_{\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}) \,. \tag{32}$$

As for the monolithic scheme, the explicit AC scheme is subject to a first-order CFL restriction on the time step for its stability. This condition will be investigated numerically in Section 4.

Remark 2 (Initial pressure). As is classical with the AC technique, an initial pressure needs to be specified. In all the numerical experiments reported in this work, the initial pressure is indeed known. If this were not the case, a possibility proposed in [45] (see also [36, Remarks 74.4 and 75.7]) is to recover the initial pressure by solving the following steady problem:

$$\begin{cases} \Delta p^0 = \underline{\nabla} \cdot \underline{f}^0 \,, \\ \dfrac{\partial p^0}{\partial \underline{n}}|_{\partial\Omega} = \Big(\underline{f}^0 - (-\nu \underline{\Delta}\, \underline{u}^0 + (\underline{u}^0 \cdot \underline{\nabla})\underline{u}^0) \Big) |_{\partial\Omega} \cdot \underline{n}_{\partial\Omega} \,. \end{cases} \tag{33}$$

Obviously, the initial pressure is zero is the fluid is initially at rest and the source term is divergence-free and has a zero normal component on the boundary.

3.2 Second-order schemes

3.2.1 Monolithic scheme

One standard technique is to use a second-order backward differentiation formula (BDF2) to devise second-order time-stepping schemes within the monolithic approach. In general, BDF2 is employed for every time step $n \geq 2$, and an implicit Euler step can still be considered for $n = 1$. With an implicit treatment of the convection term by means of a Picard algorithm, the second-order monolithic scheme reads as follows. For all $n = 2, \ldots, N$, iterate on $k \geq 1$ until convergence: Find $(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, p_{\mathrm{h}}^{n,k}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^{n,k} - 4\underline{u}_{\mathrm{C}}^{n-1} + \underline{u}_{\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) \\ \qquad + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n,k}) = l^n(\underline{v}_{\mathrm{C}}) \,, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{34}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Recall that $\widehat{\underline{u}}_{\mathrm{h}}^{n-1}$ (resp. $\widehat{\underline{u}}_{\mathrm{h}}^{n-2}$) denotes the solution given by the Picard algorithm at the time step $n-1$ (resp. $n-2$). The initialization of the iterative procedure is done with an approximation of the solution, for instance $\widehat{\underline{u}}_{\mathrm{h}}^{n-1}$ (as for the first-order scheme), or by using the second-order extrapolation formula $(2\widehat{\underline{u}}_{\mathrm{h}}^{n-1} - \widehat{\underline{u}}_{\mathrm{h}}^{n-2})$. Finally, for the BDF2 time discretization, it is possible to derive by means of classical algebraic manipulations (detailed, e.g., in [36, Lemma 68.1] for the heat equation) a time-discrete kinetic energy balance with dissipation, in the same spirit as in (26).

With an explicit treatment of the convection term, the second-order monolithic scheme reads as follows: For all $n = 2, \ldots, N$, find $(\widehat{\underline{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^{n} - 4\underline{u}_{\mathrm{C}}^{n-1} + \underline{u}_{\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n}) \\ \qquad = l^n(\underline{v}_{\mathrm{C}}) - \big(2t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n-1}, \widehat{\underline{v}}_{\mathrm{h}}) - t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-2}; \widehat{\underline{u}}_{\mathrm{h}}^{n-2}, \widehat{\underline{v}}_{\mathrm{h}})\big) \,, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{35}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Notice that a second-order extrapolation formula is used for the convective term on the right-hand side; here, we applied the extrapolation formula to the operator, but it is also possible to consider applying the extrapolation formula to the discrete velocity and then form the convection operator. Finally, as for the first-order time discretization, a dissipative kinetic energy balance is not available with an explicit treatment of the convection term, and a CFL restriction on the time step is required for stability.

3.2.2 AC scheme

It has been shown in [45] that arbitrary order in time can be achieved by combining the AC method with a bootstrap technique. In order to obtain the k-th order, k linear systems similar to (3) have to be solved. An alternative technique to reach arbitrary order in time, which is also discussed in [45], is based on a defect correction procedure, leading to a similar computational cost. To avoid the proliferation of variants, we focus only on the bootstrap technique.

As motivated in the introduction, we only consider an explicit treatment of the convection term. The second-order AC time-stepping scheme reads as follows: For all $n \geq 1$, find $(\underline{\widehat{u}}^n_{1,\mathrm{h}}, p^n_{1,\mathrm{h}}) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that for all $\underline{\widehat{v}}_\mathrm{h} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$,

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}^n_{1,\mathrm{C}} - \underline{u}^{n-1}_{1,\mathrm{C}}, \underline{v}_\mathrm{C}) + \nu(a_\mathrm{h}(\underline{\widehat{u}}^n_{1,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h}) + \eta d_\mathrm{h}(\underline{\widehat{u}}^n_{1,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h})) \\ \qquad\qquad = l^n(\underline{v}_\mathrm{C}) - b_\mathrm{h}(\underline{\widehat{v}}_\mathrm{h}, p^{n-1}_{1,\mathrm{h}}) - t_\mathrm{h}(\underline{\widehat{u}}^{n-1}_{1,\mathrm{h}}; \underline{\widehat{u}}^{n-1}_{1,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h}) \,, \end{cases} \tag{36a}$$

$$p^n_{1,\mathrm{h}} = p^{n-1}_{1,\mathrm{h}} - \nu\eta \mathrm{D}_\mathrm{h}(\underline{\widehat{u}}^n_{1,\mathrm{h}}), \quad \delta p^n_{1,\mathrm{h}} \coloneqq p^n_{1,\mathrm{h}} - p^{n-1}_{1,\mathrm{h}} \,. \tag{36b}$$

Then, for all $n \geq 2$, find $(\underline{\widehat{u}}^n_{2,\mathrm{h}}, p^n_{2,\mathrm{h}}) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that for all $\underline{\widehat{v}}_\mathrm{h} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$,

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}^n_{2,\mathrm{C}} - 4\underline{u}^{n-1}_{2,\mathrm{C}} + \underline{u}^{n-2}_{2,\mathrm{C}}, \underline{v}_\mathrm{C}) + \nu(a_\mathrm{h}(\underline{\widehat{u}}^n_{2,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h}) + \eta d_\mathrm{h}(\underline{\widehat{u}}^n_{2,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h})) \\ \quad = l^n(\underline{v}_\mathrm{C}) - b_\mathrm{h}(\underline{\widehat{v}}_\mathrm{h}, p^{n-1}_{2,\mathrm{h}} + \delta p^n_{1,\mathrm{h}}) - (2t_\mathrm{h}(\underline{\widehat{u}}^{n-1}_{2,\mathrm{h}}; \underline{\widehat{u}}^{n-1}_{2,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h}) - t_\mathrm{h}(\underline{\widehat{u}}^{n-2}_{2,\mathrm{h}}; \underline{\widehat{u}}^{n-2}_{2,\mathrm{h}}, \underline{\widehat{v}}_\mathrm{h})) \,, \end{cases} \tag{36c}$$

$$p^n_{2,\mathrm{h}} = p^{n-1}_{2,\mathrm{h}} + \delta p^n_{1,\mathrm{h}} - \nu\eta \mathrm{D}_\mathrm{h}(\underline{\widehat{u}}^n_{2,\mathrm{h}}) \,, \tag{36d}$$

with the following initialization choice: $\underline{\widehat{u}}^1_{2,\mathrm{h}} \coloneqq \underline{\widehat{u}}^1_{1,\mathrm{h}}$, $\underline{\widehat{u}}^0_{2,\mathrm{h}} \coloneqq \underline{\widehat{u}}^0_{1,\mathrm{h}} \coloneqq \underline{\widehat{u}}^0_\mathrm{h}$ for the velocity and $p^1_{2,\mathrm{h}} \coloneqq p^1_{1,\mathrm{h}}$ for the pressure.

4 Numerical results

In this section, we present numerical results to assess the performances of the AC method together with the CDO-Fb discretization. We first consider the 2D Taylor–Green vortex test case with the goal to verify the convergence rates in time for the first- and second-order time-stepping schemes and both treatments of the convection term. We also use this test case to study the CFL restriction on the time step in the case of an explicit treatment of the convection term. Then, we consider a modified 3D Taylor–Green vortex to compare all of the above strategies in terms of efficiency, that is, we compare the reached error thresholds on the velocity and the pressure with the needed CPU time. All the results are computed with the CDO module available in the open-source, single-phase CFD solver *Code_Saturne* [3]. For the 2D test cases, the linear systems are solved with a sparse direct solver available in the MUMPS [2] library. For the 3D test cases, iterative solvers are considered and further discussed in Section 4.2.

4.1 2D Taylor–Green vortex

The Taylor–Green vortex [55] is a well-known 2D test case usually considered to evaluate the performances of an unsteady Navier–Stokes solver. In fact, it constitutes an analytic solution of the 2D Navier–Stokes given by

$$\begin{cases} \underline{u}_\mathrm{TGV}(x,y) \coloneqq \exp(-2\nu t) \begin{bmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{bmatrix} , \\ p_\mathrm{TGV}(x,y) \coloneqq \dfrac{1}{4}\exp(-4\nu t)(\cos(2x) + \cos(2y)) \,. \end{cases} \tag{37}$$

The computational domain is $\Omega \coloneqq [0,\, 2\pi]^2$, whereas the observation time T depends on the numerical experiment. The viscosity ν is uniform but different values are considered in order to modify the Reynolds number Re. Since the reference length and velocity for this test case can be set to $L \coloneqq 1$ and $U \coloneqq 1$, we obtain $Re = \frac{1}{\nu}$. Cartesian and polygonal meshes are considered.

4.1.1 Convergence results

The convergence in time is first verified for all the schemes presented in Section 3. Let $\underline{v}(t,\underline{x})$ (resp. $q(t,\underline{x})$) be the analytical velocity (resp. pressure), and $\{\widehat{\underline{v}}_{\mathrm{h}}^n\}_{n=1,\dots,N} \in [\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}]^N$ (resp. $\{q_{\mathrm{h}}^n\}_{n=1,\dots,N} \in [\mathcal{P}_{\mathrm{h},*}]^N$) its space-time discretization. The following space-time error norms are used:

$$
\begin{aligned}
\|\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v}) - \{\widehat{\underline{v}}_{\mathrm{h}}^n\}_n\|_{\ell^2(L^2)}^2 &\coloneqq \sum_{n=1}^N \Delta t\, \|\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v}(t^n,\cdot)) - \widehat{\underline{v}}_{\mathrm{h}}^n\|_{\mathrm{C}}^2 \coloneqq \sum_{n=1}^N \Delta t \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, |\underline{\pi}_{\mathrm{c}}(\underline{v}(t^n,\cdot)) - \widehat{\underline{v}}_{\mathrm{c}}^n|^2 \ , \\
\|\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v}) - \{\widehat{\underline{v}}_{\mathrm{h}}^n\}_n\|_{\ell^2(H^1)}^2 &\coloneqq \sum_{n=1}^N \Delta t \sum_{\mathrm{c}\in\mathrm{C}} \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathfrak{p}_{\mathrm{f,c}}| \left| \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v}(t^n,\cdot)) - \widehat{\underline{v}}_{\mathrm{c}}^n)|_{\mathfrak{p}_{\mathrm{f,c}}} \right|^2 \ , \\
\|\underline{\pi}_{\mathrm{h}}(q) - \{q_{\mathrm{h}}^n\}_n\|_{\ell^2(L^2)}^2 &\coloneqq \sum_{n=1}^N \Delta t\, \|\pi_{\mathrm{h}}(q(t^n,\cdot)) - q_{\mathrm{h}}^n\|_{\mathrm{C}}^2 \coloneqq \sum_{n=1}^N \Delta t \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, (\underline{\pi}_{\mathrm{c}}(q(t^n,\cdot)) - q_{\mathrm{c}}^n)^2 \ .
\end{aligned}
\tag{38}
$$

These errors are then normalized by the value obtained for the corresponding norm of the exact solution.

Figure 2 shows the results for first- (top row) and second-order time-stepping schemes (bottom row) on a Cartesian mesh composed of 512^2 cells. The Reynolds number is first set to $Re = 1$ (i.e., we take $\nu = 1$), and the AC parameter is set to $\eta = 10$. We consider both the implicit and the explicit treatments of the convection term, and observe that in the present setting, no instability is observed for the considered values of the time step. This favorable situation is attributed to the moderate value of the Reynolds number, and we will see below (see Section 4.1.2) that a CFL stability restriction needs to be enforced for larger Reynolds numbers. The results presented in Figure 2 show that the monolithic and AC approaches lead to very similar errors, both for the velocity and the pressure. The convection treatment has an effect only on the pressure error (right column): the implicit treatment (squares) is more accurate than the explicit one (circles). Indeed, due to the peculiar construction of the Taylor–Green vortex for which the pressure gradient and the convection term balance out, a good approximation of the convection term leads also to an accurate approximation of the pressure.

Additional computations with higher Reynolds numbers are performed to study the dependence of the results delivered by the AC method on the parameter η. The results are reported in Table 1. As expected, the predictions of the AC method get closer to those of the monolithic approach when larger values of η are used. However, using larger values also leads to higher computational costs because it affects the conditioning of the linear system through the presence of the grad-div term. The results altogether indicate that an appropriate choice is to set $\eta = 10Re$. Indeed, this choice leads to a satisfactory trade-off between the accuracy and the conditioning of the linear systems.

4.1.2 CFL conditions with explicit convection

We investigate numerically the stability of the monolithic and AC approaches when an explicit treatment of the convection term is used. For these tests, we consider (i) three Reynolds numbers, $Re \in \{200, 500, 700\}$, (ii) an observation time T such that $T\,Re = 10^4$, (iii) $\eta = 10\,Re$ whenever the AC method is considered. We flag a computation as having diverged if, for some $n \geq 1$, we have

$$
\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^n) > 1.1\, \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^0) = 1.1\, \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{u}_0)) \,, \tag{39}
$$

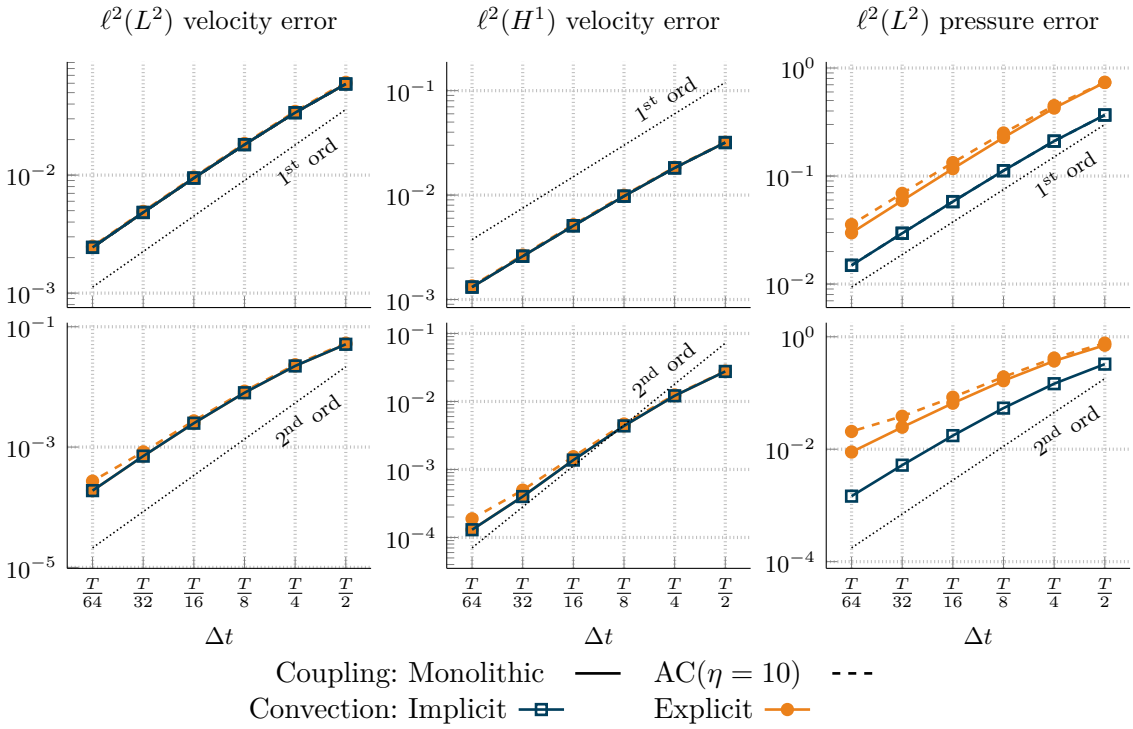

Figure 2: 2D Taylor–Green vortex. Convergence results for the space-time velocity and pressure errors. $Re = 1$, $T = 1.2$. Top: first-order schemes; bottom: second-order schemes. Cartesian mesh composed of 512^2 cells.

Table 1: 2D Taylor–Green vortex. Space-time velocity and pressure errors. Explicit convection. AC method with $\eta \in \{Re, 10Re, 100Re\}$. The errors obtained with the monolithic approach are given in parenthesis as reference. First group of columns: $Re \approx 33$, Cartesian mesh composed of 128^2 cells, $T = 40$, $\Delta t = \frac{T}{64} = 0.625$. Second group of columns: $Re = 100$, Cartesian mesh composed of 512^2 cells, $T = 120$, $\Delta t = \frac{T}{64} = 1.875$.

	$Re \approx 33$				$Re = 100$			
η	Re	$10Re$	$100Re$	(MONO)	Re	$10Re$	$100Re$	(MONO)
$\ell^2(L^2)$ velocity	1.05e−2	1.41e−3	1.24e−3	(1.24e−3)	2.91e−2	3.92e−3	1.38e−3	(1.14e−3)
$\ell^2(H^1)$ velocity	1.26e−2	1.04e−2	1.03e−2	(1.03e−2)	2.01e−2	8.25e−3	7.83e−3	(7.80e−3)
$\ell^2(L^2)$ pressure	2.81e−2	1.19e−2	1.03e−2	(1.00e−2)	6.02e−2	1.69e−2	1.25e−2	(1.19e−2)

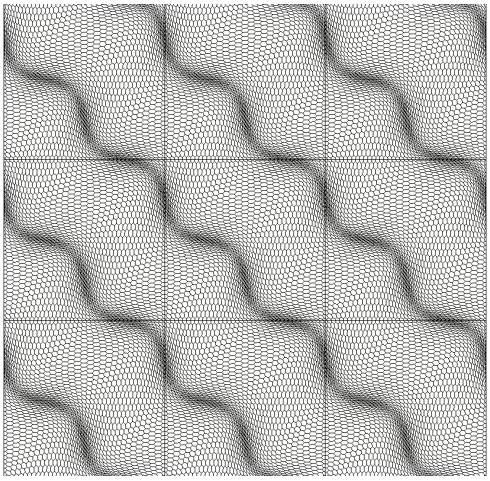

Figure 3: 2D polygonal mesh used for CFL study.

where the definition of the discrete kinetic energy $\mathcal{E}_{\mathrm{kin,h}}(\cdot)$ is given in (24). Notice that the solution (37) goes exponentially towards 0 with respect to time. Thus, a failure to satisfy (39) is a symptom of stability issues. We denote by Δt_{s} the critical time-step, that is, the largest Δt ensuring that the computation does not diverge. We seek a resolution of 1%, meaning that the gap between Δt_{s} and the smallest Δt leading to divergence is less than 1% of Δt_{s}. Whenever $\Delta t > \Delta t_{\mathrm{s}}$, we define the divergence time, T_{d}, as the smallest t^n for which (39) is satisfied.

Two meshes have been considered, a Cartesian mesh and a polygonal mesh (illustrated in Fig. 3), which have, respectively, 16,384 and 15,129 cells. In Fig. 4, we plot the divergence time, T_{d}, for different time-step values. We observe that the divergence time sharply increases as the time step is decreased towards the critical value. This confirms that using an observation time such that $T\,Re = 10^4$ is reasonable to flag stable/unstable computations. Figure 5 shows the critical time-step values for the various time-stepping schemes and the two meshes. For each case, a dependence of the critical time-step on the inverse of the Reynolds number is observed. For convenience, the results are also reported in Tables 2 and 3 for the Cartesian and the polygonal meshes, respectively, including a quantitative comparison between the first- and the second-order schemes. Generally speaking, the AC method turns out to be as stable as the monolithic scheme, and no significant differences on the critical time-step are observed. Moreover, the first-order schemes are more stable than the second-order ones, and allow one to choose time-step values more than two times larger than those possible with second-order schemes. Finally, the use of a polygonal mesh only marginally affects the value of the critical time-step.

4.2 3D modified Taylor–Green vortex

We consider the 3D modified Taylor–Green vortex solution of [19, Benchmark case 2.2] and adapt it so that it is a solution to the Navier–Stokes equations. In particular, we make the solution time-dependent by considering a sinusoidal amplitude. The exact solution reads as

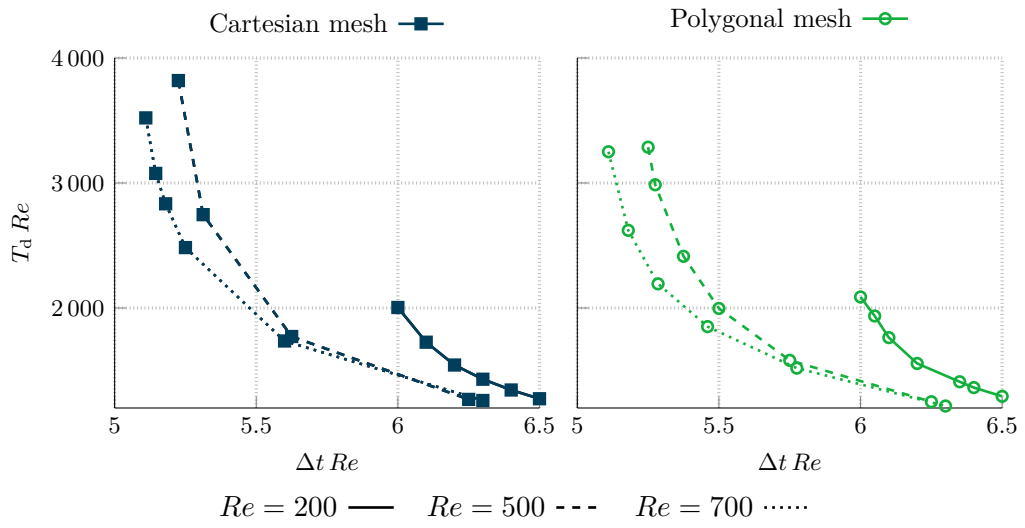

Figure 4: 2D Taylor–Green vortex. Monolithic approach and explicit convection. Divergence time T_d (scaled by Re), for different choices of Δt (scaled by Re). Left: Cartesian mesh composed of 16,384 cells. Right: polygonal mesh composed of 15,129 cells.

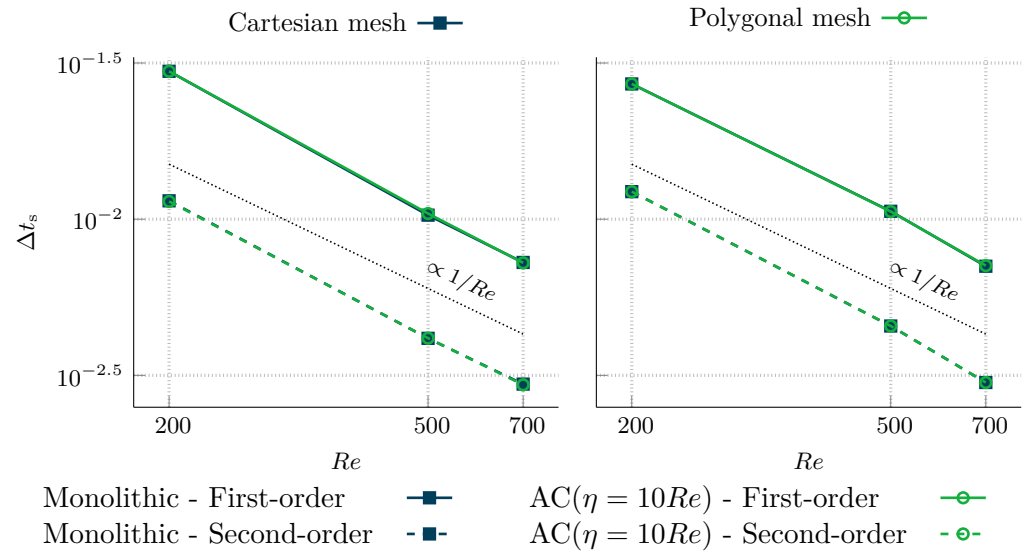

Figure 5: 2D Taylor–Green vortex. Critical time-step for stability, Δt_s (up to 1% resolution), as a function of the Reynolds number, Re, for several time-stepping schemes. Left: Cartesian mesh composed of 16,384 cells. Right: polygonal mesh composed of 15,129 cells.

Table 2: 2D Taylor–Green vortex. Critical time-step for stability, Δt_s, as a function of the Reynolds number, Re. Comparison of first- and second-order schemes. Cartesian mesh composed of 16,384 cells.

	Monolithic			AC($\eta = 10Re$)		
Re	1^{st}	2^{nd}	$\frac{1^{st}}{2^{nd}}$	1^{st}	2^{nd}	$\frac{1^{st}}{2^{nd}}$
200	$2.98e{-2}$	$1.15e{-2}$	2.60	$2.98e{-2}$	$1.14e{-2}$	2.61
500	$1.03e{-2}$	$4.16e{-3}$	2.48	$1.04e{-2}$	$4.16e{-3}$	2.51
700	$7.27e{-3}$	$2.97e{-3}$	2.45	$7.25e{-3}$	$2.95e{-3}$	2.46

Table 3: 2D Taylor–Green vortex. Critical time-step for stability, Δt_{s}, as a function of the Reynolds number, Re. Comparison of first- and second-order schemes. Polygonal mesh composed of 15,129 cells.

	Monolithic			AC($\eta = 10Re$)		
Re	1^{st}	2^{nd}	$\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	1^{st}	2^{nd}	$\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
200	$2.71e{-2}$	$1.23e{-2}$	2.21	$2.71e{-2}$	$1.23e{-2}$	2.21
500	$1.06e{-2}$	$4.55e{-3}$	2.33	$1.06e{-2}$	$4.55e{-3}$	2.33
700	$7.07e{-3}$	$3.00e{-3}$	2.36	$7.10e{-3}$	$3.00e{-3}$	2.37

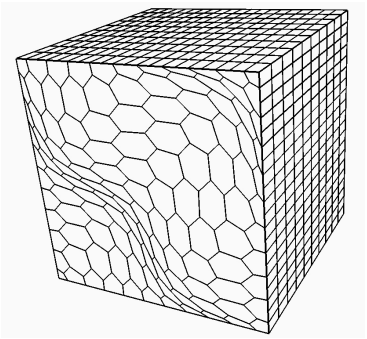

Figure 6: Coarsest mesh of the PrG sequence of [41] composed of prisms with polygonal basis.

follows:

$$\begin{cases} \underline{u}(x,y,z) \coloneqq \alpha(t)\underline{u}'(x,y,z)\,, \\ p(x,y,z) \coloneqq \alpha(t)p'(x,y,z)\,, \\ \alpha(t) \coloneqq \sin(8\pi t)\,, \\ \underline{u}'(x,y,z) \coloneqq \begin{bmatrix} -2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\sin(2\pi y)\cos(2\pi z) \end{bmatrix}, \\ p'(x,y,z) \coloneqq -6\pi\sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\,, \end{cases} \tag{40}$$

and the resulting source term is such that

$$\begin{cases} \underline{f}(x,y,z) \coloneqq \alpha(t)\underline{f}'(x,y,z) + 8\,\pi\cos(8\,\pi t)\underline{u}'(x,y,z) + \\ \qquad\qquad - \dfrac{\alpha^2(t)}{4}\begin{bmatrix} -2\sin(4\pi x)(\cos(4\pi y)+\cos(4\pi z)-2) \\ \sin(4\pi y)(\cos(4\pi x)-2\cos(4\pi z)+1) \\ \sin(4\pi z)(\cos(4\pi x)-2\cos(4\pi y)+1) \end{bmatrix}, \\ \underline{f}'(x,y,z) \coloneqq [-36\pi^2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z),\ 0,\ 0]^T\,. \end{cases} \tag{41}$$

The domain is the unit cube $\Omega \coloneqq (0,1)^3$ and the viscosity is $\nu \coloneqq 1$. Hence, since it is again reasonable to set $L \coloneqq 1$ and $U \coloneqq 1$ for the reference length and velocity, respectively, we obtain a Reynolds number of $Re = 1$. The observation time is set to $T \coloneqq 2$ and the considered time-step values are $\Delta t \in \{\frac{T}{32}, \frac{T}{64}, \frac{T}{128}, \frac{T}{256}\}$. The convection term is always treated explicitly. A quite refined mesh composed of prisms with polygonal bases is considered. Namely, the mesh obtained from the PrG sequence proposed in the benchmark [41] by gluing six of the finest meshes side by side and then rescaling the resulting mesh to the desired dimension (an example is shown in Fig. 6). The resulting mesh has more than 14M cells and leads to a final coupled system of more than 170M DoFs (after static condensation).

Only iterative solvers are considered in this 3D test case. In particular, the linear systems

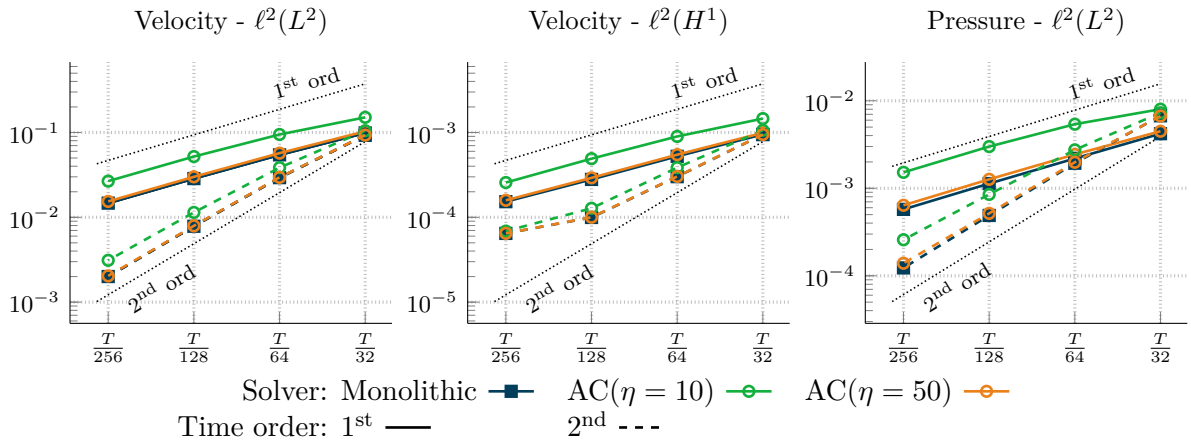

Figure 7: 3D modified Taylor–Green vortex. Convergence results for the space-time velocity and pressure errors. $Re = 1$.

obtained with the AC method are solved by means of a Jacobi-preconditioned Conjugate Gradient (CG) solver. In order to deal with the saddle point systems obtained with the monolithic approach, an iterative Golub-Kahan Bidiagonalization (GKB) [4, 5] procedure without augmentation is chosen; the internal solver is a Conjugate Gradient with a K-cycle Algebraic Multigrdid preconditioner [54]. All these solvers are natively available in *Code_Saturne*. The computations have been run on the GAIA cluster of EDF[1] on up to 700 cores. Two values of the AC parameter have been considered: $\eta = 10Re$ and $\eta = 50Re$.

Fig. 7 reports the convergence results for the normalized space-time velocity and pressure errors as defined in (38). These results corroborate the optimal convergence in time for both first- and second-order schemes. A slight decrease of the convergence rate for the velocity $\ell^2(H^1)$-error is observed for the finest Δt and the second-order schemes, which indicates that the space discretization error tends to become dominant. Moreover, we remark that the higher the value of η, the more accurate the results obtained with the AC method. Actually, for $\eta = 50Re$, the errors are essentially superimposed to those obtained with the monolithic approach.

Our aim is now to compare the computational efficiency of the monolithic and AC methods by studying the achieved velocity and pressure errors as a function of the computational cost (defined as the product between the elapsed time and the number of computing cores). As a preliminary step, computations are run in order to find reasonably optimized values for the tolerances of the iterative solvers. The results for the first-order AC and monolithic approaches are, respectively, gathered in Tables 4 and 5. A tolerance of 10^{-4} appears to be a reasonable choice for the CG iterative solver in the AC context, for both considered values of η. In order to anticipate the greater accuracy of second-order schemes, the tolerance is decreased to 10^{-5} for the second-order AC time-stepping scheme. Similarly, for the monolithic approach, the tolerances for the GKB and CG iterative solvers are both set to 10^{-4} and 10^{-5}, respectively, for the first- and second-order time-stepping schemes.

We can now proceed to the assessment of the computational efficiency of the various schemes with the tolerances prescribed as above. The performances of the three strategies (monolithic, AC($\eta = 10$), and AC($\eta = 50$)), for both first- and second-order schemes, are compared in Fig. 8. The AC method turns out to be more efficient: having fixed the order in time for the three compared methods, the AC($\eta = 10$) and AC($\eta = 50$) methods can achieve a given error threshold (y-axis) in less computation time (x-axis) than the monolithic method. The performance of AC depends on the parameter η: the higher η, the more numerical effort (hence,

[1] 243rd of the TOP500 list of November 2020

Table 4: 3D modified Taylor–Green vortex. Space-time errors for different tolerances of the CG linear solver. First-order AC time-stepping scheme, $\Delta t = \frac{T}{128}$. The adopted tolerances for further calculations are circled.

ϵ(CG)	$\ell^2(L^2)$ $\underline{u}$	$\ell^2(H^1)$ $\underline{u}$	$\ell^2(L^2)$ p
		$\eta = 10$	
$1e{-}2$	$6.58e{-}2$	$6.76e{-}4$	$3.17e{-}3$
$\boxed{1e{-}4}$	$5.20e{-}2$	$4.91e{-}4$	$3.00e{-}3$
$1e{-}6$	$5.20e{-}2$	$4.91e{-}4$	$3.00e{-}3$
		$\eta = 50$	
$1e{-}2$	$3.83e{-}2$	$4.22e{-}4$	$1.49e{-}3$
$\boxed{1e{-}4}$	$3.00e{-}2$	$2.92e{-}4$	$1.27e{-}3$
$1e{-}6$	$3.00e{-}2$	$2.92e{-}4$	$1.27e{-}3$

Table 5: 3D modified Taylor–Green vortex. Space-time errors for different tolerances of the GKB and CG linear solvers. First-order monolithic time-stepping scheme, $\Delta t = \frac{T}{128}$. The adopted tolerances for further calculations are circled.

ϵ(GKB)	ϵ(CG)	$\ell^2(L^2)$ $\underline{u}$	$\ell^2(H^1)$ $\underline{u}$	$\ell^2(L^2)$ p
$1e{-}2$	$1e{-}2$	$4.27e{-}2$	$8.31e{-}4$	$2.80e{-}3$
	$1e{-}4$	$2.84e{-}2$	$3.21e{-}4$	$1.28e{-}3$
	$1e{-}6$	$2.83e{-}2$	$3.20e{-}4$	$1.28e{-}3$
$\boxed{1e{-}4}$	$\boxed{1e{-}4}$	$2.86e{-}2$	$2.80e{-}4$	$1.13e{-}3$
	$1e{-}6$	$2.87e{-}2$	$2.80e{-}4$	$1.13e{-}3$
$1e{-}6$	$1e{-}4$	$2.86e{-}2$	$2.80e{-}4$	$1.13e{-}3$
	$1e{-}6$	$2.87e{-}2$	$2.80e{-}4$	$1.13e{-}3$

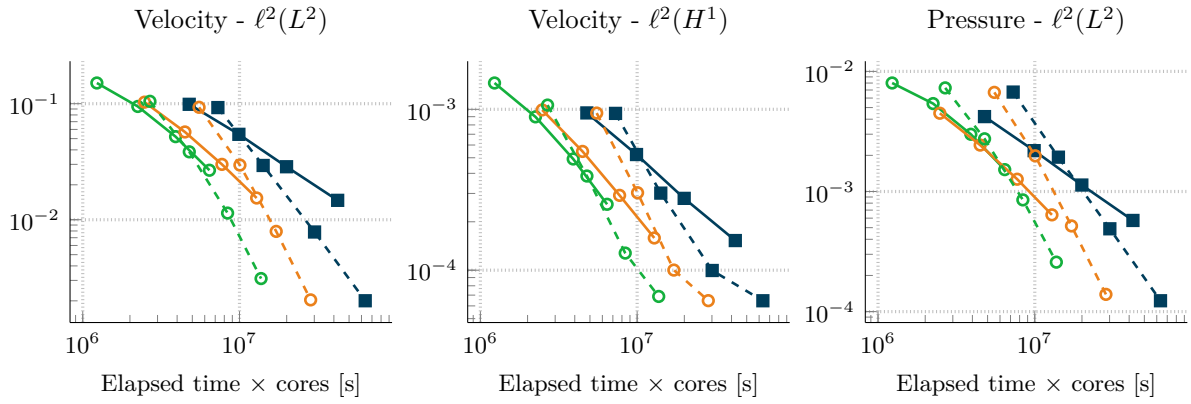

Figure 8: 3D modified Taylor–Green vortex. Comparison of computational efficiency for the various first- and second-order schemes. Legend: see Fig. 7

Table 6: 3D modified Taylor–Green vortex. Comparison of accuracy and performance for the various first- and second-order schemes. Time step set to $\Delta t = \frac{T}{128}$

	$\left\|\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})\right\|_{\ell^2,\mathrm{C}}$			$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\ell^2,\mathrm{C}}$		
Solver	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
Monolithic	$2.86e{-2}$	$7.84e{-3}$	3.6	$2.80e{-4}$	$9.96e{-5}$	2.8
AC(10)	$5.20e{-2}$	$1.14e{-2}$	4.6	$4.91e{-4}$	$1.28e{-4}$	3.8
AC(50)	$3.00e{-2}$	$7.93e{-3}$	3.8	$2.92e{-4}$	$1.00e{-4}$	2.9
	$\|\mathrm{E}_{\mathrm{h}}(p)\|_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
Solver	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
Monolithic	$1.13e{-3}$	$4.89e{-4}$	2.3	$2.00e{+7}$	$3.01e{+7}$	0.7
AC(10)	$3.00e{-3}$	$8.51e{-4}$	3.5	$3.91e{+6}$	$8.39e{+6}$	0.5
AC(50)	$1.27e{-3}$	$5.17e{-4}$	2.4	$7.73e{+6}$	$1.71e{+7}$	0.5

computational time) is required, and altogether the choice $\eta = 10$ turns out to be more effective than the choice $\eta = 50$. When considering the second-order schemes, the difference between the AC and monolithic approaches is somewhat mitigated. One explanation is that the bootstrap procedure employed in the second-order AC requires two linear system resolutions per time step, whereas only one is needed with the BDF2 and monolithic coupling. Finally, the results show an advantage of the second-order schemes over the first-order ones, especially when accurate calculations are considered. A comparison at the fixed time-step $\Delta t = \frac{T}{128}$ (corresponding to the finest time step before the space discretization errors tend to become influential, see the discussion of Fig. 7) is presented in Table 6. Focusing first on the bottom right part of the table related to the computational times, one can compare the performances of monolithic and AC methods, with the latter being consistently two times faster than the former, while still providing satisfactory errors. Moreover, the errors obtained with second-order schemes are more than two times smaller than those obtained with first-order schemes, while the computation time are less than two times larger.

5 Conclusions

We have extended the CDO-Fb schemes to the unsteady incompressible Navier–Stokes problem and investigated the accuracy and efficiency of the artificial compressibility (AC) technique for the time discretization using either a first-order time-stepping scheme or a second-order one. The assessment of the AC technique is made by means of systematic comparisons with the standard monolithic approach, using either a first-order or a second-order version. First, optimal convergence rates in time have been recovered for all the time-stepping schemes in various norms for the velocity and the pressure. Moreover, the stability of the time-stepping schemes when the nonlinear convection term is treated explicitly has been investigated on Cartesian and polygonal meshes, showing in both cases a linear dependency on the reciprocal of the Reynolds number, with only a slightly tighter restriction when using polygonal meshes. Finally, an assessment of the performance of all the schemes has been carried out using large 3D polytopal meshes. The AC method proved to be an efficient alternative to the classical monolithic approach: it ensures accuracy levels close to those obtained with the monolithic approach while using only half of the computational time. Moreover, a comparison between first- and second-order time-stepping schemes indicates an advantage for the latter: second-order schemes provide errors as much as four times smaller than those obtained with first-order schemes, while taking only up to twice the computation time.

As an outlook, the improvement of the preconditioning for large linear systems arising from

the AC technique is paramount to achieve scalable and higher efficiency since the presence of the $\underline{\nabla}\,\underline{\nabla}\cdot$ term prevents the usual preconditioners such as algebraic multigrid techniques to perform well. The construction of a specific preconditioner adapted from the seminal work in [6] along with the more recent ones in [11] and [39] will be the subject of further investigations.

Acknowledgments

The PhD fellowship of R. Milani was partially supported by EDF R&D and ANRT.

References

[1] J. Aghili, S. Boyaval, and D. A. Di Pietro, *Hybridization of mixed high-order methods on general meshes and application to the Stokes equations*, Comput. Methods Appl. Math., 15 (2015), pp. 111–134.

[2] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.

[3] F. Archambeau, N. Méchitoua, and M. Sakiz, *Code Saturne: A Finite Volume Code for Turbulent flows - Industrial Applications*, Int. J. Finite Vol., 1 (2004).

[4] M. Arioli, *Generalized Golub–Kahan bidiagonalizationand stopping criteria*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 571–592.

[5] M. Arioli, C. Kruse, R. Ulrich, and N. Tardieu, *An iterative generalized Golub-Kahan algorithm for problems in structural mechanics*, Tech. Rep. August, 2018.

[6] D. N. Arnold, R. S. Falk, and R. Winther, *Preconditioning in H(div) and applications*, Math. Comp., 66 (1997), pp. 957–984.

[7] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267–279.

[8] L. Beirão da Veiga, V. Gyrya, K. Lipnikov, and G. Manzini, *Mimetic Finite Difference method for the Stokes problem on polygonal meshes*, J. Comput. Phys., 228 (2009), pp. 7215–7232.

[9] L. Beirão da Veiga, C. Lovadina, and G. Vacca, *Divergence free Virtual Elements for the Stokes problem on polygonal meshes*, ESAIM Math. Model. Numer. Anal., 51 (2017), pp. 509–535.

[10] ——, *Virtual Elements for the Navier-Stokes problem on polygonal meshes*, SIAM J. Numer. Anal., 56 (2018), pp. 1210–1242.

[11] M. Benzi, M. A. Olshanskii, and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, Int. J. Numer. Meth. Fluids, 66 (2011), pp. 486–508.

[12] J. Bonelle, *Compatible Discrete Operator schemes on polyhedral meshes for elliptic and Stokes equations*, PhD thesis, Université Paris-Est, 2014.

[13] J. Bonelle, D. A. Di Pietro, and A. Ern, *Low-order reconstruction operators on polyhedral meshes: application to compatible discrete operator schemes*, Comput. Aided Geom. Des., 35 (2015), pp. 27–41.

[14] J. Bonelle and A. Ern, *Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 553–581.

[15] ——, *Analysis of Compatible Discrete Operator schemes for the Stokes equations on polyhedral meshes*, IMA J. Numer. Anal., 35 (2015), pp. 1672–1697.

[16] J. Bonelle, A. Ern, and R. Milani, *Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations*, in Finite Vol. Complex Appl. IX; Methods Theor. Aspects, R. Klöfkorn, E. Keilegavlen, F. A. Radu, and J. Fuhrmann, eds., vol. 323 of Springer Proc. Math. Stat., Bergen, 2020, Springer International Publishing, pp. 93–101.

[17] L. Botti, D. A. Di Pietro, and J. Droniou, *A Hybrid High-Order method for the incompressible Navier–Stokes equations based on Temam's device*, J. Comput. Phys., 376 (2019), pp. 786–816.

[18] F. Boyer, S. Krell, and F. Nabet, *Inf-Sup stability of the discrete duality finite volume method for the 2D Stokes problem*, Math. Comput., 84 (2015), pp. 2705–2742.

[19] C. Cancès and P. Omnes, eds., *Finite Volumes for Complex Applications VIII - Methods and Theoretical Aspects*, vol. 199 of Springer Proc. Math. Stat., Lille, France, 6 2017, Springer International Publishing.

[20] A. Cangiani, V. Gyrya, and G. Manzini, *The nonconforming Virtual Element Method for the Stokes equations*, SIAM J. Numer. Anal., 54 (2016), pp. 3411–3435.

[21] P. Cantin, *Approximation of scalar and vector transport problems on polyhedral meshes*, PhD thesis, Université Paris-Est, 2016.

[22] P. Cantin and A. Ern, *An edge-based scheme on polyhedral meshes for vector advection-reaction equations*, ESAIM Math. Model. Numer. Anal., 51 (2017), pp. 1561–1581.

[23] A. J. Chorin, *Numerical Solution of the Navier–Stokes Equations*, Math. Comput., 22 (1968), pp. 745–762.

[24] B. Cockburn, G. Kanschat, D. Schötzau, and C. Schwab, *Local discontinuous Galerkin methods for the Stokes system*, SIAM J. Numer. Anal., 40 (2002), pp. 319–343.

[25] B. Cockburn, N. C. Nguyen, and J. Peraire, *A Comparison of HDG Methods for Stokes Flow*, J. Sci. Comput., 45 (2010), pp. 215–237.

[26] S. Delcourte, *Développement de méthodes de volumes finis pour la mécanique des fluides*, PhD thesis, Université Paul Sabatier, 2007.

[27] D. A. Di Pietro, J. Droniou, and A. Ern, *A discontinuous-skeletal method for advection-diffusion-reaction on general meshes*, SIAM J. Numer. Anal., 53 (2015), pp. 2135–2157.

[28] D. A. Di Pietro and A. Ern, *A Hybrid High-Order locking-free method for linear elasticity on general meshes*, Comput. Methods Appl. Mech. Eng., 283 (2015), pp. 1–21.

[29] D. A. Di Pietro, A. Ern, A. Linke, and F. Schieweck, *A discontinuous skeletal method for the viscosity-dependent Stokes problem*, Comput. Methods Appl. Mech. Eng., 306 (2016), pp. 175–195.

[30] D. A. Di Pietro and S. Krell, *A Hybrid High-Order method for the steady incompressible Navier–Stokes problem*, J. Sci. Comput., 74 (2018), pp. 1677–1705.

[31] D. A. Di Pietro and S. Lemaire, *An extension of the Crouzeix-Raviart space to general meshes with application to quasi-incompressible linear elasticity and Stokes flow*, Math. Comput., 84 (2015), pp. 1–31.

[32] J. Droniou and R. Eymard, *Study of the mixed finite volume method for Stokes and Navier–Stokes equations*, Numer. Methods Partial Differ. Equ., 25 (2009), pp. 137–171.

[33] ——, *Benchmark: Two hybrid mimetic mixed schemes for the lid-driven cavity*, in Finite Vol. Complex Appl. VIII; Methods Theor. Aspects, vol. 199 of Springer Proc. Math. Stat., Lille, 2017, Springer International Publishing, pp. 107–124.

[34] J. Droniou, R. Eymard, and P. Feron, *Gradient Schemes for Stokes problem*, IMA J. Numer. Anal., 36 (2015), pp. 1636–1669.

[35] J. Droniou, R. Eymard, T. Gallouët, and R. Herbin, *Gradient schemes: a generic framework for the discretisation of linear, nonlinear and nonlocal elliptic and parabolic equations*, Math. Model. Methods Appl. Sci., 23 (2013), pp. 2395–2432.

[36] A. Ern and J.-L. Guermond, *Finite Elements III*, vol. 74 of Texts in Applied Mathematics, Springer International Publishing, 2021.

[37] R. Eymard, P. Feron, and C. Guichard, *Family of convergent numerical schemes for the incompressible Navier–Stokes equations*, Math. Comput. Simul., 144 (2018), pp. 196–218.

[38] R. Eymard, T. Gallouët, and R. Herbin, *Discretisation of heterogeneous and anisotropic diffusion problems on general nonconforming meshes. SUSHI: a scheme using stabilisation and hybrid interfaces*, IMA J. Numer. Anal., 30 (2010), pp. 1009–1043.

[39] P. E. Farrell, L. Mitchell, and F. Wechsung, *An Augmented Lagrangian preconditioner for the 3D stationary incompressible Navier–Stokes equations at high Reynolds number*, SIAM J. Sci. Comput., 41 (2019), pp. 3037–3096.

[40] P. Feron, *Gradient Schemes for some elliptic and parabolic, linear and non-linear problems*, PhD thesis, université Paris-Est, 2016.

[41] J. Fořt, J. Fürst, J. Halama, R. Herbin, and F. Hubert, eds., *Finite Volumes for Complex Applications VI - Problems & Perspectives*, vol. 4 of Springer Proc. Math. Stat., Prague, Czech Republic, 6 2011, Springer Science & Business Media.

[42] G. N. Gatica, M. Munar, and F. A. Sequeira, *A mixed Virtual Element Method for the Navier–Stokes equations*, Math. Model. Methods Appl. Sci., 28 (2018), pp. 2863–2904.

[43] T. Goudon, S. Krell, and G. Lissoni, *DDFV method for Navier–Stokes problem with outflow boundary conditions*, Numer. Math., 142 (2019), pp. 55–102.

[44] J.-L. Guermond and P. Minev, *High-order adaptive time stepping for the incompressible Navier-Stokes equations*, SIAM J. Sci. Comput., 41 (2019), pp. A770–A788.

[45] J.-L. Guermond and P. D. Minev, *High-order time stepping for the incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 37 (2015), pp. A2656–A2681.

[46] P. Hansbo and M. G. Larson, *Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 1895–1908.

[47] Y. Jeon, E.-J. Park, and D. Sheen, *A hybridized finite element method for the Stokes problem*, Computers and Mathematics with Applications, 68 (2014), pp. 2222–2232.

[48] S. Krell and G. Manzini, *The Discrete Duality Finite Volume method for Stokes equations on three-dimensional polyhedral meshes*, SIAM J. Numer. Anal., 50 (2012), pp. 808–837.

[49] O. A. Ladyzhenskaya, *The mathematical theory of viscous incompressible flow*, vol. 2, Gordon and Breach, New York, NY, 1969.

[50] C. Lehrenfeld and J. Schöberl, *High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows*, Comput. Methods Appl. Mech. Engrg., 307 (2016), pp. 339–361.

[51] R. Milani, *Compatible Discrete Operator schemes for the unsteady incompressible Navier–Stokes equations*, PhD thesis, Université Paris-Est, 2020.

[52] L. Mu, X. Wang, and X. Ye, *A modified weak Galerkin finite element method for the Stokes equations*, J. Comput. Phys., 275 (2015), pp. 79–90.

[53] N. C. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations*, J. Comput. Phys., 230 (2011), pp. 1147–1170.

[54] Y. Notay and P. S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numer. Linear Algebra Appl., 15 (2008), pp. 473–487.

[55] G. I. Taylor and A. E. Green, *Mechanism of the production of small eddies from large ones*, Proc. R. Soc. Lond. A, 158 (1937), pp. 499–521.

[56] R. Temam, *Sur l'Approximation de la Solution des Equations de Navier–Stokes par la Méthode des Pas Fractionnaires (I)*, Arch. Ration. Mech. Anlysis, 32 (1969), pp. 135–153.

[57] N. Vladimirova, B. Kuznetsov, and N. N. Yanenko, *Numerical calculation of the symmetrical flow of viscous incompressible liquid around a plate*, Some Problems in Computational and Applied Mathematics, Nauka, (1966).

[58] N. N. Yanenko, *The method of fractional steps*, Springer-Verlag, New York, NY, 1971.